Homework 2: Queues

1. A keyboard buffer on a computer’s operating system is implemented as a circular queue.

(a) Explain why a circular queue is an appropriate data structure choice. [2]

(b) A particular keyboard buffer consists of five cells in a circular queue. The queue **kBuffer** is initialised by setting a variable **size** (containing the number of items in the array) to 0, pointers **front** to 0 and **rear** to -1. A variable **maxSize** holds the maximum size of the queue.

(i) Complete the table to show the results after the following operations. [4]

			kBuffer						
	[0]	[1]	[2]	[3]	[4]		size	front	rear
Initial state							0	0	-1
Enqueue S									
Enqueue W									
Dequeue									
Enqueue E									

(ii) Complete the table to show the results after the following operations. [3]

			kBuffer						
	[0]	[1]	[2]	[3]	[4]		size	front	rear
Current state	J	U	X	L	M		3	1	3
Enqueue T									
Enqueue R									
Dequeue									

(c) Code for the keyboard buffer operations needs to be written.

Use the variables defined in part (b): kBuffer, maxSize, size, front, and rear.

(i) Write the pseudocode for the isFull() operation, including function header. [2]

(ii) Write the pseudocode for the deQueue operation. [4]

(d) (i) Describe, with the aid of an example, the operation of a priority queue from the user's point of view. [2]

(ii) Explain how the principles of data abstraction and encapsulation can be used to hide the details of implementation of a priority queue. [3]

Total 20 marks